NASA Contractor Report 181978
ICASE Report No. 90-4

# ICASE



## AN ANALYSIS OF SCATTER DECOMPOSITION

David M. Nicol
Joel H. Saltz

Contract Nos. NAS1-18107, NAS1-18605
January 1990

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665-5225

Operated by the Universities Space Research Association



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

# An Analysis of Scatter Decomposition

*David M. Nicol**
*The College of William and Mary*
*Joel H. Saltz*†
*Institute for Computer Applications in Science and Engineering*



**Abstract**

This paper provides a formal analysis of a powerful mapping technique known as scatter decomposition. Scatter decomposition divides an irregular computational domain into a large number of equal sized pieces, and distributes them modularly among processors. We use a probabilistic model of workload in one dimension to formally explain why, and when scatter decomposition works. Our first result is that if correlation in workload is a convex function of distance, then scattering a more finely decomposed domain yields a lower average processor workload variance. Our second result shows that if the workload process is stationary Gaussian and the correlation function decreases linearly in distance until becoming zero and then remains zero, scattering a more finely decomposed domain yields a lower expected maximum processor workload. Finally we show that if the correlation function decreases linearly across the entire domain, then among all mappings that assign an equal number of domain pieces to each processor, scatter decomposition minimizes the average processor workload variance. The dependence of these results on the assumption of decreasing correlation is illustrated with situations where a coarser granularity actually achieves better load balance.

*This research was supported in part by NASA contracts NAS1-18107 and NAS1-18605, and NSF Grant ASC 8819373.
†Supported in part by NASA contracts NAS1-18107 and NAS1-18605, the Office of Naval Research under contract No. N000[illegible]-86-K-0310, and NSF grant DCR 8106181.

# 1 Introduction

Scatter decomposition [1], (also described as modular mapping [4]) is an effective method for parallelizing a large class of irregular scientific programs that are tied to physical domains. Examples include a wide variety of techniques for numerically solving time dependent partial differential equations, and other, less numerical domain-oriented simulations. Scatter decomposition divides the domain into a set of rectangular regions with the same spatial size and geometry. The regions are labeled using Cartesian coordinates, and are mapped to processors by applying the mod function to the label in each coordinate. For example, Figure 1 shows how a two dimensional irregular grid for a PDE is decomposed into strips (marked by the heavy lines) and assigned to processors. The execution of all workload related to a subregion is a basic unit of schedulable work which we call a *cluster*. A cluster's granularity is controlled by the parameters defining the region size, in this case the strip width.

Scatter decomposition's success lies in its ability to balance workload without ever actually analyzing it. Any region of high workload tends to be subdivided and distributed among processors. Scatter decomposition is a technique applied to many problems in many contexts [1, 2, 4, 5, 9, 11, 14, 17]. Its success has been explained informally in [1] and [4], by appealing to the physics and numerics of many scientific computations. While these explanations suffice for most practitioners, the literature lacks a full formal analysis of why scatter decomposition balances workload. This paper provides some such analysis, identifying model assumptions under which scatter decomposition can be expected to effectively balance load. As such, our work is a necessary prerequisite for any future formal treatment of the very important problem of managing the inherent tensions between load imbalance and communication costs in a scatter decomposition.

The object of this paper is to construct and analyze a performance model to explain when and why scatter decomposition works. The model is based on a number of simplifying assumptions to promote tractability. As such, it should not be viewed as a model that accurately *predicts* performance quantitatively. Rather, it should be viewed as a model that *explains* performance qualitatively. Specifically, we model workload in a one dimensional domain as a continuous second-order stationary process. This means that we associate a random workload with every point in the domain, assume that the mean workload at every point is the same, assume that the workload variance at every point is the same, and assume that the covariance between the workloads at any two points is uniquely determined by their distance. The model takes the domain to be divided into some $n = 2^d$ clusters of equal size, mapped modularly onto $P = 2^p$ processors. Throughout this paper we take $P$ to be fixed, and $d \geq p$. The *degree* of the decomposition is defined to be $d$. Given one scatter decomposition, another of higher degree can be constructed by splitting each cluster into two, then by modularly mapping the resulting set of clusters.

We derive three main results, each of which has a different set of assumptions concerning the correlation function.

1. *Assumption:* The correlation function is convex. *Result:* Increasing the degree of a scatter decomposition does not increase the common processor workload variance.

2. *Assumptions.* The workload process is stationary and Gaussian. The correlation function decreases linearly until reaching zero, then remains zero (an elbow function). *Result:* There exists a degree $d_0$, such that if $d_0 \leq d_1 < d_2$, then the expected maximum processor workload under a scatter decomposition of degree $d_2$ is no larger than the expected maximum processor workload under a scatter decomposition of degree $d_1$.

3. *Assumption.* The correlation function decreases linearly across the entire domain. *Result:* For any number of clusters $2^l$, among all mappings that assign $2^{l-p}$ clusters per processor the modular mapping minimizes the average processor workload variance.

Performance ultimately is measured in terms of finishing time, so that the expected load of the most heavily loaded processor is an appropriate metric. One of our results addresses this metric directly. Average processor workload variance is a secondary measure, although intuition does suggest that decreasing the variance while keeping the mean constant will decrease the expected maximum. Consequently, all these results confirm our intuition that modularly mapping increasingly finer grained workload leads to better load balance. It should be noted that increased communication overhead is the price paid for this balance, and is a cost we do not include in this model. One should not interpret these results as saying that better overall performance can always be achieved by increasing the degree. For a given domain, there will be an optimal degree that balances the conflicting goals of low communication costs and good load balance.

A brief analysis of scatter decomposition can be found in [15]. However, that analysis assumes statistical independence between all cluster workloads, and seems to consider the effects of scatter decomposition on a given architecture as the problem size is increased. As such it is an inappropriate model for studying the effects of changing the mapping of a single given problem. Treatments of other problems have used stochastic models of workload to estimate the expected finishing time, but invariably those models concern statistically independent workloads, e.g. the analyses in [3] and [6]. These results are inadequate for analyzing scatter decomposition. When all workload is independent, then aggregated workload is independent, and there is no performance benefit to be gained by scattering. Scatter decomposition is successful precisely because the workload is not independent. Our contribution is to propose and analyze a model that includes workload correlation, and explain why increasingly finer partitions mapped modularly tend to balance the load better.

## 2 Analysis

In this section we study a probabilistic model of workload, and the performance of different mappings. For the sake of simplicity we constrain our model to be one-dimensional. This assumption does not negate the utility of the model; any multi-dimensional problem partitioned into hyper-strips can be viewed as a one-dimensional problem. Such partitions greatly simplify the programming needed to exchange information between processors. In fact, our experience in mapping a land-battle simulation using scatter decomposition was that strip partitions minimized the execution time [10]. This was also our experience in mapping a regular scientific code onto the Intel iPSC/1 [16].

Our analysis concerns the effect of scatter decomposition on load balance, in the absence of communication or synchronization costs. By understanding how load balance in isolation is affected by the decomposition/mapping decisions, we are better able to understand the tension between load imbalance and communication/synchronization overheads. The model we use is intended to be descriptive, rather than predictive; the analysis is qualitative rather than quantitative. We doubt that the end benefits of fitting a model to performance data will justify the costs of doing so. Nevertheless we feel there is worth in formally affirming the intuition behind scatter decomposition.

### 2.1 When and Why Scatter Decomposition Works

Our model explains the success of scatter decomposition by showing that it induces correlation between processors' workloads. To see the performance benefits of correlated workloads, imagine that a random workload is generated and partitioned so that the same amount of work is assigned to every processor. A processor's workload is random, but all processors always finish at the same time, because their workloads are perfectly correlated. This situation is optimal, because all processors are busy all the time. Now imagine that the workload at every point is statistically independent of any other. No matter what the domain decomposition or mapping, processor workloads are statistically independent. In fact, the expected maximum processor workload is the same regardless of granularity, so long as the same volume of domain is assigned to each processor. The "ideal" of random but highly correlated processor workloads cannot be achieved in this artificial scenario.

Scatter decomposition works because irregular workloads are *not* statistically independent: high workload tends to appear in contiguous regions. A sufficiently fine-grained decomposition will split the region up, modular assignment will spread its workload around. The contribution of that region to one processor's workload is highly correlated with the contribution of a nearby region to a different processor's workload. If the underlying workload is highly correlated in nearby regions, then scatter decomposition induces correlation between processors' workloads. We have observed this phenomenon in our own experiments with a one-

dimensional fluid flow computation using adaptive gridding [11]. The fluids problem exhibits irregular grids similar to those in Figure 1.

For a given problem, the sample autocorrelation function [12](p. 437) is a statistical estimate of correlation between point workloads, as a function of the distance between them. Autocorrelations range between 1 and -1; the larger the autocorrelation, the more similar the workloads of two points at a given distance tend to be. Zero correlation implies statistical independence; increasingly negative correlations imply increasing dissimilarity between workloads. Figure 2 shows the sample autocorrelation function at one time-step in a fluid flow computation. Not only does correlation diminish as a function of distance, it can reasonably be modeled as a convex "elbow" function $d_a(t) = \sigma^2 \max\{0, 1 - at\}$ over an appropriate range of $t$, and some $a \geq 0$. This corresponds nicely with two of our results, one of which assumes elbow correlation, the other of which assumes a convex correlation function.

There are situations where scatter decomposition will not work well. Consider a one dimensional domain discretized into 1000 points, numbered between 0 and 999, to be mapped onto ten processors. Randomly choose some "base" number $b \in [0, 99]$, and imagine that every hundredth point beginning with $b$ has a computational cost of 1000, while all other points have a computational cost of 1. If one evenly divides the domain into ten subregions and maps them modularly, every processor has 1099 units of computation to execute. Scatter a decomposition of twenty subregions, and half the processors each have a computational cost of 2098, while the other half each have a cost of 100. Modularly assign each point individually, and processor ($b$ mod 10) has a cost of 10090, while every other processor has a cost of 100. In this situation mapping increasingly finer-grained workload leads to decreasing performance. Due to $b$'s randomness this workload model is stochastic, and is second-order stationary. Two points at a distance $100m$ for $m = 1, \ldots, 9$ will always have the same workload. The correlation function at all distances $100m$ consequently has value one. It has some fixed smaller value for all other distances. The principle reason this problem defeats fine-grained scatter decomposition is the periodicity. One should be extremely careful using scatter decomposition in the presence of strong periodic behavior, if there is any chance that the periodicity of the modular mapping can align with the periodicity of workload. The assumptions of the models we study do not admit periodicity.

## 2.2 Model Preliminaries

We consider the behavior of a computation over a real line interval, divided into $n$ clusters, and mapped onto $P$ processors. Both $n$ and $P$ are taken to be powers of two, and $n \geq P$. We are interested in the average processor workload variance, and in the expected workload of the processor that takes the longest time to complete. Without loss of generality we take the real interval to be $[0, 1]$. Assume that every point $p \in [0, 1]$ has a certain *work intensity* $W(t)$. The time required to process $[a, b]$ is the integral of $W(t)$ from $t = a$ to $t = b$. We assume that the intensities $W(t)$ are unknown, but we are willing to model our uncertainity by

assuming that $W(t)$ is a random variable, and that $W(t)$ can be viewed as a second-order stationary process [13] over $t \in [0,1]$. Thus we suppose that $E[W(t)] = \mu$ for all $t \in [0,1]$, that $Var[W(t)] = \sigma^2$ for all $t \in [0,1]$, and that $Cov[W(t), W(s)]$ depends only on $|t - s|$. To emphasize this point we will denote the covariance function as $Cov(|t-s|)$. These assumptions are reasonable if we are unwilling or unable to differentiate between the likely behavior of the computation at $t$ and at $s$. We do *not* assume that $W(t) = W(s)$, we simply assume that we have the same degree of uncertainty about $W(t)$ and $W(s)$.

The execution time for $[a,b]$ is

$$T(a,b) = \int_a^b W(t)\,dt.$$

$T(a,b)$ has mean value $(b-a)\mu$. The variance of $T(a,b)$ is

$$\begin{aligned} Var[T(a,b)] &= E[T(a,b)^2] - (b-a)^2\mu^2 \\ &= E\left[\left(\int_a^b W(t)\,dt\right)\left(\int_a^b W(s)\,ds\right)\right] - (b-a)^2\mu^2 \\ &= \int_a^b \int_a^b E[W(t)W(s)]\,dt\,ds \;-\; (b-a)^2\mu^2 \\ &= \int_a^b \int_a^b Cov(|s-t|)\,dt\,ds \;-\; (b-a)^2\mu^2. \end{aligned} \tag{1}$$

Following a decomposition into $n$ clusters, $i$'th cluster's workload is $T(i/n,(i+1)/n)$, and is denoted as $c_i(n)$. The random vector of cluster workloads is denoted $C(n) = <c_0(n), \ldots, c_{n-1}(n)>$.

We are interested in the covariance matrix $\sigma_C^2$ for the cluster workloads. For $i \neq j$ we have

$$Cov[c_i(n), c_j(n)] = (\sigma_C^2)_{ij} = \int_{i/n}^{(i+1)/n} \int_{j/n}^{(j+1)/n} Cov(t-s)\,dt\,ds. \tag{2}$$

$Var[c_i(n)]$ is simply $Var[T(i/n,(i+1)/n)]$, given above. The sequence $c_0(n), c_1(n), \ldots, c_{n-1}(n)$ is second-order stationary, a fact easily deduced from equations (1) and (2). To emphasize this we define the function $\phi$:

$$\phi(|j-i|, n) = Cov[c_i(n), c_j(n)].$$

Note that $\phi(0,n)$ is a cluster's variance.

An assignment of clusters to processors is described by a $P \times n$ *assignment* matrix whose *ij-th* entry is 1 if $c_j(n)$ is assigned to processor $i$, and is 0 otherwise. Given assignment matrix $\mathcal{A}$, the multiplication $\mathcal{A}C$ yields a $P \times 1$ random vector whose *jth* component is the sum of the execution times of all clusters assigned to processor $j$. The vector of mean processor loads is the matrix-vector product $\mathcal{A}\bar{\mu}_n$, where $\bar{\mu}_n$ is the $n$ element vector with $\mu/n$ in each coordinate. The covariance matrix of $\mathcal{A}C$ is the product $\mathcal{A}\sigma_C^2\mathcal{A}^T$, where $\mathcal{A}^T$ is the transpose of $\mathcal{A}$. The overall execution time is the maximum processor execution time, or $\max\{(\mathcal{A}C)^T\}$. This quantity is random.

For any processor $P_i$, let $\mathcal{A}(i)$ denote the set of clusters assigned to it under $\mathcal{A}$, and let $L_i(\mathcal{A}, n)$ be $P_i$'s random workload. By definition the variance of $L_i(\mathcal{A}, n)$ is given by

$$\begin{aligned} Var[L_i(\mathcal{A}, n)] &= (\mathcal{A}\sigma_C^2\mathcal{A}^T)_{ii} \\ &= \sum_{c_j(n)\in\mathcal{A}(i)} \phi(0, n) + \sum_{<c_j(n),c_k(n)>\in\mathcal{A}(i)\times\mathcal{A}(i)} \phi(|j-k|, n). \end{aligned} \tag{3}$$

The first component of this expression is the sum of variances of all clusters assigned to $P_i$. The second component is a sum of *cluster covariance terms* (we will call these cc terms), that depends on the assignment. Similarly, the covariance between processors $L_i(\mathcal{A}, n)$ and $L_j(\mathcal{A}, n)$ is given by a sum of cc terms:

$$Cov[L_i(\mathcal{A}, n), L_j(\mathcal{A}, n)] = \sum_{<c_k(n),c_m(n)>\in\mathcal{A}(i)\times\mathcal{A}(j)} \phi(|k-m|, n) \tag{4}$$

The sum of all cluster covariance matrix terms always equals the sum of all processor workload variances and covariances[1]

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(\sigma_C^2)_{ij} = \sum_{i=0}^{P-1}\sum_{j=0}^{P-1}(\mathcal{A}\sigma_C^2\mathcal{A}^T)_{ij}.$$

This implies a balance between processor workload variances and covariances (and hence correlations); if by changing $\mathcal{A}$ we reduce the average processor workload variance, then we are increasing the average inter-processor workload correlation.

The indices of the sums (3) and (4) have special structure when $\mathcal{A}$ describes a modular mapping. We know that if $c_j(n)$ and $c_k(n)$ are assigned to the same processor, then $|j-k|$ is a multiple of $P$. Under a modular mapping each processor will have $n/P$ clusters. Among these there are $n/P-1$ pairs of clusters whose indices are exactly $P$ apart, $n/P-2$ pairs whose indices are exactly $2P$ apart, and so on. Since $n$ and $P$ determine the specifics of the mapping we may drop the notational dependence of $L_i(\mathcal{A}, n)$ on $\mathcal{A}$. Under the modular mapping we may write the common processor workload variance as

$$Var[L(n)] = (n/P)\phi(0, n) + 2\sum_{k=1}^{(n/P)-1}((n/P)-k)\phi(kP, n). \tag{5}$$

To consider processor workload covariance under a modular assignment take $i < j$, and consider a cluster $c_a(n)$ assigned to processor $P_i$. It has cc terms with all processor $P_j$ clusters $c_m(n)$ such that $|a-m| \bmod P = j-i$ or $|a-m| \bmod P = P-j+i$. There are $((n/P)-k)$ cc terms arising from clusters whose indices are $kP+j-i$ apart (for $k = 0, \ldots, (n/P)-1$); there are $((n/P)-k)$ cc terms arising from clusters whose indices are $kP-j+i$ apart (for $k = 1, \ldots, (n/P)-1$). We may therefore write

$$Cov[L_i(n), L_j(n)] = \sum_{k=0}^{(n/P)-1}((n/P)-k)\phi(kP+j-i, n) + \sum_{k=1}^{(n/P)-1}((n/P)-k)\phi(kP-j+i, n)$$

[1] this conservation law proved to be invaluable when debugging detailed expressions for the processor workload variance and covariances, e.g. (12) and (13).

$$= (n/P)\phi(j-i,n) + \sum_{k=1}^{(n/P)-1} ((n/P)-k)\phi(kP+j-i,n) + \sum_{k=1}^{(n/P)-1} ((n/P)-k)\phi(kP-j+i,n). \quad (6)$$

## 2.3 Decreasing Workload Variance

Under very general assumptions one can show that increasing the degree of a scatter decomposition reduces the common processor workload variance. The necessary assumptions are that the workload process be second-order stationary, and that its covariance function be convex.

The first step is to show that $\phi(|j-i|,n)$ is a convex function of $|j-i|$ over the range $1,2,\ldots,n-1$. Towards this end assume that $x \geq 1/n$ and define

$$\begin{aligned} I(n,x) &= E\left[\int_0^{1/n}\int_x^{x+1/n} W(s)W(t)\,dt\,ds\right] \\ &= \int_0^{1/n}\int_x^{x+1/n} Cov(t-s)\,dt\,ds \\ &= \int_0^{1/n}\left[\int_x^{\infty} Cov(t-s)\,dt - \int_{x+1/n}^{\infty} Cov(t-s)\,dt\right] ds \end{aligned}$$

Taking the derivative with respect to $x$ we find that

$$\frac{\partial}{\partial x} I(n,x) = \int_0^{1/n} (Cov(x+1/n-s) - Cov(x-s))\,ds.$$

The difference being integrated increases in $x$ due to $Cov(t)$ convexity, implying that the derivative of $I(n,x)$ with respect to $x$ increases in $x$—one characterization of a convex function. By stationarity $Cov[c_i(n),c_j(n)] = Cov[c_0(n),c_{|j-i|}(n)]$; furthermore $Cov[c_0(n),c_{|j-i|}(n)] = I(n,|j-i|/n)$. Consequently $Cov[c_i(n),c_j(n)]$ is a convex function of $|j-i|$ once $|j-i| \geq 1$ (it may indeed be convex over the entire range, but that fact has not been shown, and is not needed).

We are interested in the effects of moving from a scatter decomposition with degree $d-1$ to one with degree $d$. To analyze these effects we make the following observation. Consider a domain partitioned into $n = 2^d$ clusters, which is mapped by modularly assigning *pairs* of clusters: $c_0(n)$ and $c_1(n)$ are assigned to processor 0, $c_2(n)$ and $c_3(n)$ are assigned to processor 1, and so on. This mapping is identical to the scatter decomposition of degree $d-1$; the pair of clusters $c_0(n), c_1(n)$ viewed from the $d$ degree mapping is the same as the single cluster $c_0(n/2)$ viewed from the $d-1$ degree mapping. We will show that the modular mapping with degree $d-1$ produces processor variances that are no smaller than those of the modular mapping with degree $d$.

Split each cluster $c_i(n/2)$ into two equal sized clusters. The sum of the two split cluster variances plus

twice their covariance must equal the variance of $c_i(n/2)$. That is,

$$\phi(0, n/2) = 2\phi(0, n) + 2\phi(1, n). \tag{7}$$

Similarly, take two clusters $c_i(n/2)$ and $c_j(n/2)$, and split each into two equal sized clusters. The total covariance between the four split clusters must equal the covariance between the two unsplit clusters. Thus

$$\phi(|j-i|, n/2) = 2\phi(2|j-i|, n) + \phi(2|j-i|+1, n) + \phi(2|j-i|-1, n). \tag{8}$$

Note that the index values must double when taken with respect to $n$ rather than $n/2$ clusters.

Substituting the right-hand-sides of equations (7) and (8) into equation (5) and working through the algebra, we find that

$$Var[L(n/2)] = (n/P)\phi(0, n) + (n/P)\phi(1, n) + 2\sum_{k=1}^{(n/(2P))-1} ((n/P) - 2k)\phi(2kP, n) + \sum_{k=1}^{(n/(2P))-1} ((n/P) - 2k)[\phi(2kP+1, n) + \phi(2kP-1, n)].$$

Using this expression and (5), we compute the difference $Var[L(n/2)] - Var[L(n)]$. All terms involving $\phi(2kP, n)$ cancel, for $k = 0, \ldots, n/(2P) - 1$. Each remaining term from $Var[L(n)]$ has the form $2((n/P) - 2k - 1)\phi((2k+1)P, n)$, for $k = 0, \ldots, n/(2P) - 1$. We split each such term into the sum $(n/P - 2k)\phi((2k+1)P, n) + (n/P - 2k - 2)\phi((2k+1)P, n)$, and pair these with $Var[L(n/2)]$ terms as follows:

$$Var[L(n/2)] - Var[L(n)] = \sum_{k=0}^{(n/(2P))-1} ((n/P - 2k)(\phi(2kP+1, n) - \phi((2k+1)P, n)) - (n/P - 2k - 2)(\phi((2k+1)P, n) - \phi((2k+2)P - 1, n))). \tag{9}$$

One characteristic of a convex function $g$ is that for fixed $y$ the difference $g(x) - g(x+y)$ is a decreasing function of $x$. Every two terms we have paired differ in their index arguments by exactly $P - 1$, e.g., $\phi(2kP+1, n)$ and $\phi((2k+1)P, n)$. Since $\phi$ is a convex function of the index argument once the index is at least 1, we have for every $k$

$$\phi(2kP+1, n) - \phi((2k+1)P, n) \geq \phi((2k+1)P, n) - \phi((2k+2)P - 1, n).$$

The left-hand-side expression in this inequality is weighted more heavily in equation (9) than is the right-hand-side expression. It follows that $Var[L(n/2)] - Var[L(n)] \geq 0$, proving our first result.

**Theorem 1** *Suppose the workload process $W(t)$ is second-order stationary with a convex covariance function. Then increasing the degree of a scatter decomposition does not increase the processor workload variance.*

□

## 2.4 Decreasing Expected Maximum Workload

Next we demonstrate circumstances where increasing the degree of a scatter decomposition reduces the expected workload of the most heavily loaded processor. The argument is to show that under appropriate assumptions the correlation between any two processors' workloads increases as the degree increases. We then cite a result from the literature proving that the expected maximum decreases in this situation.

We assume that the workload process $\{W(t)\}$ is a stationary Gaussian process[2] [7]. Additivity properties of the Gaussian then ensure that the vector of $n$ clusters has a jointly normal distribution [7](Chapter 6) and that under any assignment, the processors' workloads are jointly normal. We also assume that the correlation function is $Cov(t) = \sigma^2 \max\{0, 1 - \alpha t\}$, where $\alpha = 2^v/m_0 \geq 1$ for some integers $v, m_0 \geq 0$. The restriction on $\alpha$ is used to simplify certain calculations. $\delta = 1/\alpha$ is the smallest distance $t$ at which $Cov(t) = 0$. Our results apply when the degree is large enough so that subinterval $[0, \delta]$ is partitioned into at least $P = 2^p$ clusters. If the degree is $d$, then the number of clusters in $[0, \delta]$ is $\delta 2^d$. Now let $d_0$ be the least $d$ such that $\delta 2^d \bmod 2^p = 0$. Equivalently, $d_0$ is the least integer $d$ such that $m_0 2^{d-p-v}$ is an integer. Clearly $d_0 \leq p+v$. Our results apply when the degree is at least $d_0$.

We can compute functio..al forms for $\phi(|j-i|, n)$ given this explicit definition of $Cov(t)$. Performing the integration given by (2) one determines that

$$\phi(|j-i|, n) = \begin{cases} \frac{\sigma^2}{n^3}(n - \alpha|j-i|) & \text{if } |j-i| < \delta n \\ \frac{\sigma^2 \alpha}{6n^3} & \text{if } |j-i| = \delta n \\ 0 & \text{if } |j-i| > \delta n \end{cases} . \tag{10}$$

These calculations take advantage of the fact that $\delta$ is a multiple of $1/n$. The variance of a cluster is determined by evaluating (1), yielding

$$\phi(0, n) = \frac{\sigma^2}{n^3}(n - \alpha/3). \tag{11}$$

Given equations (10) and (11) we can compute processor workload variance and covariance under scatter decomposition. General expressions for these quantities are given by (5) and (6). For large values of $k$, some terms in those sums vanish, being zero. Our assumption that the scatter decomposition has degree $d_0$ or larger ensures that terms which vanish are easily characterized,[3] and that those clusters whose indices are exactly $\delta n$ apart are assigned to the same processor. All $\phi(kP, n)$ terms in (5) vanish for $k > \delta n/P$; we have $\phi(kP, n) = \sigma^2\alpha/(6n^3)$ for $k = \delta n/P$. We may rewrite the variance as

$$Var[L(n)] = (n/P)\frac{\sigma^2(n - \alpha/3)}{n^3} + \sum_{k=1}^{(\delta n/P)-1} \left( (n/P - k)\frac{\sigma^2(n - \alpha k P)}{n^3} \right) + (n/P - \delta n/P)\left(\frac{\sigma^2\alpha}{6n^3}\right)$$

[2] note that this assumption is stronger than we have used so far, due both to stationarity rather than second-order stationarity, and due to the assumption of a specific workload distribution

[3] This is not the case for smaller degrees. A large number of special cases must be constructed and analyzed. This task seemed to us to be more tedious than is warranted by the anticipated correspondingly stronger result.

$$= \sigma^2\left(\frac{(\delta-\delta^2/3)}{P^2}+\frac{1-\alpha/P}{3n^2}-\frac{1-\alpha}{3n^2P}\right). \tag{12}$$

Calculation of this equality is much amplified with the use of a symbolic mathematics package.

The processor workload covariance is similarly handled. Assume that $i < j$. $k = \delta n/P$ again delineates where $\phi$ terms vanish: $\phi(kP+j-i,n) = 0$ for all $k \geq \delta n/P$, and $\phi(kP-i+j,0) = 0$ for all $k > \delta n/P$. We may rewrite (6) as

$$\begin{aligned} Cov[L_i(n), L_j(n)] &= (n/P)\frac{\sigma^2}{n^3}(n-\alpha(j-i)) + \sum_{k=1}^{(\delta n/P)-1}((n/P)-k)\frac{\sigma^2(n-\alpha(kP+j-i))}{n^3} + \\ &\qquad \sum_{k=1}^{\delta n/P}((n/P)-k)\frac{\sigma^2(n-\alpha(kP-j+i))}{n^3} \\ &= \sigma^2\left(\frac{\delta-\delta^2/3}{P^2}+\frac{1}{3n^2}-\frac{j-i}{Pn^2}\right). \end{aligned} \tag{13}$$

The correlation between $L_i(n)$ and $L_j(n)$ is the ratio $Cov[L_i(n), L_j(n)]/Var[L(n)]$. For all $d \geq d_0$ we obtain the correlation using (13) and (12), and can treat the ratio as a continuous function of $n$. It is interesting to note that as $n$ increases the correlation approaches unity. This supports our intuition that partitioning the domain into increasingly finer clusters and mapping them modularly induces correlation between processor workloads. In fact, the tendency towards unity is monotonic. Taking the derivative with respect to $n$ we find that the derivative is positive if

$$(4/3-2\delta/3)(j-i)+2\delta/9-2/3 > 0.$$

This inequality holds, since $(4/3-2\delta/3) \geq 2/3$. Consequently, for all $n = 2^d \geq 2^{d_0}$ we must have

$$Cov[L_i(2n), L_j(2n)]/Var[L(2n)] > Cov[L_i(n), L_j(n)]/Var[L(n)].$$

Next we use this relationship to analyze the expected maximum processor workload.

The following result is based on the Normal Comparison Lemma [8](p.81) and is the key to our observations concerning the expected maximum processor workload.

**Theorem 2 (Leadbetter et al.)** *Let $\xi_0, \ldots, \xi_k$ be standardized jointly normal random variables, and let $\eta_0, \ldots, \eta_k$ be standardized jointly normal random variables, such that $Cov(\xi_i, \xi_j) \leq Cov(\eta_i, \eta_j)$ for each $i, j$, $i \neq j$. Then for every $u$,*

$$Pr\{\max\{\xi_0, \ldots, \xi_k\} \leq u\}\} \leq Pr\{\max\{\eta_0, \ldots, \eta_k\} \leq u\}\},$$

*and hence*

$$E[\max\{\xi_0, \ldots, \xi_k\}] \geq E[\max\{\eta_0, \ldots, \eta_k\}].$$

□

The standardization of a random variable $X$ is the scaled random variable $Z = (X - m)/s$, where $m$ and $s$ are $X$'s mean and standard deviation, respectively. The mean of a standardized random variable is zero and its variance is one; the covariance between two standardized random variables is the correlation between their corresponding unstandardized forms. Let $Z_i(n)$ be the standardized workload of processor $P_i$ given a domain of $n$ clusters. $Cov[Z_i(n), Z_j(n)] = Cov[L_i(n), L_j(n)]/Var[L(n)]$, which we have shown to be increasing in $n$. If $\tilde{n} > n$ (equivalently, if one scatter decomposition has higher degree than another), then

$$E[\max\{Z_0(n), \ldots, Z_{P-1}(n)\}] \geq E[\max\{Z_0(\tilde{n}), \ldots, Z_{P-1}(\tilde{n})\}]. \tag{14}$$

The expected maximum workload is

$$\begin{aligned} E[\max\{L_0(n), \ldots, L_{P-1}(n)\}] &= E[\max_{0 \leq i \leq P-1}\{L_i(n) + Var[L(n)]^{1/2} Z_i(n)\}] \\ &= \mu/P + Var[L(n)]^{1/2} E[\max_{0 \leq i \leq P-1}\{Z_i(n)\}]. \end{aligned}$$

Theorem 1 shows that $Var[L(n)] \geq Var[L(2n)]$; this along with inequality (14) proves our second result.

**Theorem 3** *Let $\{W(t)\}$ be a stationary Gaussian process, with a covariance function $Cov(t) = \sigma^2 \max\{0, 1 - \alpha t\}$, where $\alpha = 2^v/m_0 \geq 1$ for some positive integers $m_0$, $v$. Let there be $2^p$ processors, and let $d_0$ be the least integer $d$ such that $m_0 2^{d-p-v}$ is an integer. If $d_2 > d_1 \geq d_0$, then the expected maximum processor workload of a scatter decomposition with degree $d_2$ is no greater than that of a scatter decomposition with degree $d_1$.*

□

## 2.5 Minimization of Average Workload Variance

Our final result gives conditions where for a given $n$, among all "balanced" assignments—those placing $n/P$ clusters per processor—the modular mapping minimizes the average processor workload variance. To prove this result we assume that the covariance function decreases linearly across the entire domain: $Cov(s) = \sigma^2(1 - \alpha s)$, for some $\alpha$ satisfying $0 \leq \alpha \leq 2$. The result is based on a procedure that takes any assignment and constructs another whose sum of processor workload variances is no larger. The repeated application of this procedure produces a modular assignment. Consequently, modular assignments minimize the average processor workload variance.

The arguments to follow specify individual covariance terms. These arguments are clearer using the $Cov[c_i(n), c_j(n)]$ notation rather than $\phi(|j-i|, n)$. It is straightforward to determine the form of $Cov[c_i(n), c_j(n)]$ under the present assumptions:

$$Cov[c_i(n), c_j(n)] = \begin{cases} \frac{\sigma^2}{n^3}(n - \alpha|j-i|) & \text{if } |j-i| > 0 \\ \frac{\sigma^2}{n^3}(n - \alpha/3) & \text{if } |j-i| = 0 \end{cases}. \tag{15}$$

Let $\mathcal{A}_1$ be any assignment matrix describing a balanced assignment. Without loss of generality, we assume that under $\mathcal{A}_1$ the processors are numbered so that $P_0$ is assigned $c_0(n)$, $P_1$ is assigned the smallest indexed $c_i(n)$ that is not assigned to $P_0$, and in general $P_j$ is assigned the smallest indexed cluster that is not assigned to any of $P_0, P_1, \ldots, P_{j-1}$.

We will say that $c_j(n)$ is *in place* if it is assigned to processor $P_{j \bmod p}$. Note that all clusters are in place under a modular assignment. We construct another balanced assignment $\mathcal{A}_2$ by finding the smallest indexed $c_i(n)$ that is not in place, and by putting it in place. Let $c_f$ denote this cluster, let $P_S$ denote the *source* processor that has $c_f$ under $\mathcal{A}_1$, and let $P_T$ denote the *target* processor $P_{f \bmod p}$. Let $c_g$ be the smallest indexed cluster assigned to $P_T$ such that $g > f$. $\mathcal{A}_2$ is constructed from $\mathcal{A}_1$ by giving $c_f$ to $P_T$, and $c_g$ to $P_S$. Figure 3 illustrates these definitions. We will prove that the sum of processor variances under $\mathcal{A}_2$ bounds that sum under $\mathcal{A}_1$ from below; consequently the average workload variance under $\mathcal{A}_2$ is no greater than that under $\mathcal{A}_1$.

Recall that under any assignment matrix $\mathcal{A}$ the variance of $P_i$'s work load is given by

$$\begin{aligned} Var[L_i(\mathcal{A}, n)] &= \left(\mathcal{A}\sigma_C^2\mathcal{A}^T\right)_{ii} \\ &= \sum_{c_j(n)\in\mathcal{A}(i)} Var[c_j(n)] + \sum_{\langle c_j(n), c_k(n)\rangle \in \mathcal{A}(i)\times\mathcal{A}(i)} Cov[c_j(n), c_k(n)], \end{aligned} \tag{16}$$

and that

$$Cov[L_i(\mathcal{A}, n), L_j(\mathcal{A}, n)] = \sum_{\langle c_k, c_m\rangle \in \mathcal{A}(i)\times\mathcal{A}(j)} Cov[c_k(n), c_m(n)]$$

It is clear from (16) that the variance of any processor other than $P_S$ or $P_T$ is by unaffected by swapping $c_f$ and $c_g$. To prove the desired result we need only show that the swap does not increase the sum of $P_T$ and $P_S$ variances. The change in processor variances caused by the swap is entirely due to changes in the sum of cluster covariance (cc) terms in each processor. After swapping $c_f$ and $c_g$, each cluster $c_i(n)$ assigned to $P_S$ loses the cc term $Cov[c_f(n), c_i(n)]$ and gains the term $Cov[c_g(n), c_i(n)]$. We let $\Delta_{L_S}$ denote the sum of all such changes among clusters in $P_S$ to the left of $c_f$, and let $L_S$ denote the number of such clusters. Similarly $\Delta_{R_S}$ denotes the sum of changes among clusters in $P_S$ to the right of $c_g$ and $R_S$ denotes the number of such clusters; $\Delta_{M_S}$ denotes the sum of changes among clusters in $P_S$ with indices between $f$ and $g$. Expressions for these quantities are derived using equation (15):

$$\Delta_{L_S} = \sum_{\substack{c_i(n)\in\mathcal{A}_2(S) \\ i<f}} \left(Cov[c_g(n), c_i(n)] - Cov[c_f(n), c_i(n)]\right) = -\frac{\sigma^2}{n^3}(g-f)L_S\alpha;$$

$$\Delta_{R_S} = \sum_{\substack{c_j(n)\in\mathcal{A}_2(S) \\ j>g}} \left(Cov[c_g(n), c_j(n)] - Cov[c_f(n), c_j(n)]\right) = \frac{\sigma^2}{n^3}(g-f)R_S\alpha;$$

$$\Delta_{M_S} = \sum_{\substack{c_k(n)\in\mathcal{A}_2(S) \\ f<k<g}} \left(Cov[c_g(n), c_k(n)] - Cov[c_f(n), c_k(n)]\right) = \frac{\sigma^2}{n^3} \sum_{\substack{c_k(n)\in\mathcal{A}_2(S) \\ f<k<g}} (2k-f-g)\alpha.$$

The change in $P_S$'s variance after the swap is the sum $\Delta_{L_S} + \Delta_{M_S} + \Delta_{R_S}$.

We can similarly describe the change in $P_T$'s variance with the definitions

$$\Delta_{L_T} = \sum_{\substack{c_i(n) \in A_2(T) \\ i<f}} (Cov[c_f(n), c_i(n)] - Cov[c_g(n), c_i(n)]) = \frac{\sigma^2}{n^3}(g-f)L_T\alpha;$$

$$\Delta_{R_T} = \sum_{\substack{c_j \in A_2(T) \\ j>g}} (Cov[c_f(n), c_j(n)] - Cov[c_g(n), c_j(n)]) = -\frac{\sigma^2}{n^3}(g-f)R_T\alpha.$$

No term analogous to $\Delta_{M_S}$ is necessary since there are no clusters in $P_T$ with indices between $f$ and $g$.

The change in the sum of $P_S$'s variance with $P_T$'s variance is given by the sum of all the $\Delta$ terms defined above. We will show that the sum of $\Delta$ terms is bounded from above by 0. At this point a number of observations are useful. Since all $c_i(n)$ with $i < f$ are in order, it follows that $L_T \leq L_S$. Thus $\Delta_{L_S} + \Delta_{L_T} \leq 0$. It remains to show that $\Delta_{R_S} + \Delta_{R_T} + \Delta_{M_S} \leq 0$. We know that

$$\Delta_{R_S} + \Delta_{R_T} = -\frac{\sigma^2}{n^3}(R_T - R_S)(g-f)\alpha; \tag{17}$$

furthermore, since $n/P = L_T + R_T + 1$, we must also have $R_S \leq R_T$. We proceed to show that the magnitude of $\Delta_{M_S}$ is no greater than the magnitude of (17) and consequently prove the larger result.

$m = n/P - L_S - R_S - 1$ is the number of clusters in $P_S$ whose indices lie strictly between $f$ and $g$. $\Delta_{M_S}$ is maximized when the indices of these clusters are as large as possible; when $k = g-1, g-2, \ldots, g-m$. With such indices, the sum of $c_g$'s cc terms in $P_S$ is

$$\frac{\sigma^2}{n^3}\sum_{i=1}^{m}(n - i\cdot\alpha).$$

Likewise, the sum of $c_f$'s cc terms in $P_S$ is

$$\frac{\sigma^2}{n^3}\sum_{i=1}^{m}(n - (g-f-i)\alpha).$$

From this, we see that $\Delta_{M_S}$ when maximized can be written as

$$\Delta_{M_S} = \frac{\sigma^2}{n^3}\sum_{i=1}^{m}(n - i\cdot\alpha) - \frac{\sigma^2}{n^3}\sum_{i=1}^{m}(n - (g-f-i)\alpha) = \frac{\sigma^2}{n^3}m(g-f)\alpha.$$

But note that

$$\begin{aligned} m &= n/P - L_S - R_S - 1 \\ &\leq n/P - L_T - R_S - 1 \\ &= (n/P - L_T - R_T - 1) + (R_T - R_S) \\ &= (R_T - R_S), \end{aligned}$$

so that

$$\Delta_{R_S} + \Delta_{R_T} + \Delta_{M_S} = \frac{\sigma^2}{n^3}(-(R_T - R_S)(g-f)\alpha + m\cdot(g-f)\alpha) \leq 0.$$

Consequently, swapping $c_f$ and $c_g$ does not increase the sum of $P_S$ and $P_T$'s variance. Furthermore, the swap does not affect the sum of other processors' variances. Repeatedly applying this procedure puts every cluster in place, which is the modular assignment. This discussion has proved the following theorem.

Theorem 4 *Let $\{W(t)\}$ be a second-order stationary process, with a covariance function $Cov(s) = \sigma^2(1 - \alpha s)$, where $0 \leq \alpha \leq 2$. Let $P$ and $n$ be given such that $P$ divides $n$ evenly, and let $\mathcal{A}_M$ be the $P \times n$ assignment matrix describing the modular mapping. Then for any $P \times n$ assignment matrix $\mathcal{A}$ describing a balanced assignment,*

$$(1/P)\sum_{i=0}^{P-1}(\mathcal{A}_M\sigma_C^2\mathcal{A}_M{}^T)_{ii} \leq (1/P)\sum_{i=0}^{P-1}(\mathcal{A}\sigma_C^2\mathcal{A}^T)_{ii}.$$

□

In the event that the workload process is Gaussian and stationary, we can show that increasing the degree reduces the expected maximum processor workload. We determine the processor variance and covariance under scatter decomposition by substituting the values given by (15) into (5) and (6). Assume that $i < j$. Working through the algebra one determines that

$$Var[L(n)] = \sigma^2\left(\frac{1-\alpha/3}{p^2} + \frac{\alpha(1-1/P)}{3n^2}\right),$$

and that

$$Cov[L_i(n), L_j(n)] = \sigma^2\left(\frac{1-\alpha/3}{p^2} + \frac{\alpha}{3n^2} - \frac{(j-i)\alpha}{Pn^2}\right).$$

The derivative with respect to $n$ of $C[L_i(n), L_j(n)]/Var[L(n)]$ is positive if

$$(4/3 - 2\alpha/3)(j-i) + 2\alpha/9 - 2/3 > 0.$$

This is always true over the range $\alpha \in [0, 2]$. Consequently the same arguments used to prove Theorem 3 can be applied here.

# 3 Summary

Scatter decomposition is an attractive method for mapping domain-oriented computations with irregular workloads to parallel architectures. Scatter decomposition partitions the domain into $n$ equal-size pieces, and maps them modularly onto $P$ processors. This paper uses a formal probabilistic model of correlated workload in a one-dimensional domain to explain why and when scatter decomposition works. First, we show that periodicity in workload correlation can lead to load imbalance under scatter decomposition if the

correlation period aligns with the period of the modular mapping. Consequently we consider nonperiodic workload correlation functions.

Our first result shows that if workload correlation is a convex function of distance, then scattering with increasingly finer grained clusters decreases a processor's workload variance, thereby increasing the average inter-processor workload correlation. Since the processor workload mean is unaffected by this change, one anticipates that the expected maximum workload will correspondingly decrease.

Our second result affirms this intuition under a stronger set of assumptions: the workload process is Gaussian, and the correlation function decreases linearly in distance until it reaches zero and then stays at zero. We then show that once a scatter decomposition is sufficiently fine-grained, making the grain-size finer reduces the expected maximum processor workload.

Our third result shows that under slightly different assumptions still, among all possible "balanced" mappings scatter decomposition minimizes the average processor workload variance. This result depends on the correlation function decreasing linearly across the entire domain. In this case it is also true that if the workload process is Gaussian, then scattering a finer-grained decomposition reduces the expected maximum processor workload.

These analytic results serve to formally verify the intuition behind scatter decomposition. However, the results only concern load balance. The additional communication cost of decreasing granularity is not built into this model. Extensions to this work might find the optimal granularity by determining a quantitative estimator of the expected maximum workload and the expected communication cost as a function of granularity. An overall execution time model would be constructed depending on the influence of architecture on the communication costs, and then optimized.

# References


[1] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker. *Solving Problems on Concurrent Computers.* Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[2] G. A. Geist and M. T. Heath. Matrix factorization on a hypercube multiprocessor. In *The Proceedings of the Hypercube Microprocessors Conf., Knoxville, TN*, pages 161–180, September 1986.

[3] P. Heidelberger. Discrete-event simulations and parallel processing: statistical properties. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1114–1132, November 1988.

[4] T. Hoshino. *Pax Computer.* Addison-Wesley, New York, 1989.

[5] I. Ipsen, Y. Saad, and M.H. Schultz. Complexity of dense linear system solution on a multiprocessor ring. *Lin. Algebra Appl.*, 77:205–239, 1986.

[6] C. P. Kruscal and A. Weiss. Allocating independent subtasks on parallel processors. *IEEE Trans. on Soft. Eng.*, SE-11(10):1001–1015, October 1985.

[7] H.J. Larson and B.O. Shubert. *Probabilistic Models in Engineering Sciences.* Volume 1, Wiley, New York, 1979.

[8] M.R. Leadbetter, G. Lindgren, and H. Rootzén. *Extremes and Related Properties of Random Sequences and Processes.* Springer-Verlag, New York, 1983.

[9] R. Morison and S.W. Otto. The scattered decomposition for finite element problems. *Journal of Scientific Computing*, 2(1):59–76, March 1987.

[10] D.M. Nicol. Mapping a battlefield simulation onto parallel message-passing architectures. In *Proceedings of the 1988 SCS Conference on Distributed Simulation*, pages 141–146, February 1988.

[11] D.M. Nicol and P.F Reynolds, Jr. Optimal dynamic remapping of parallel computations. IEEE Trans. on Computers, 1990. To appear.

[12] R. S. Pindyck and D. L. Rubinfeld. *Econometric Models and Economic Forecasts.* McGraw-Hill, New York, 1976.

[13] H.S. Ross. *Stochastic Processes.* Wiley, New York, 1983.

[14] Y. Saad. Communication complexity of the gaussian elimination algorithm on multiprocessors. *Lin. Algebra Appl.*, 77:315 –340, 1986.

[15] J. Salmon. A mathematical analysis of the scattered decomposition. In *The Third Conference on Hypercube Concurrent Computers and Applications, Vol. 1*, pages 239–240, ACM Press, 1988.

[16] J.H. Saltz, V. K. Naik, and D.M. Nicol. Reduction of the effects of the communication delays in scientific algorithms on message passing MIMD architectures. *SIAM J. Sci. Stat. Comput*, 8(1):s118, 1987.

[17] Y. Won and S. Sahni. Maze routing on a hypercube multiprocessor computer. In *Proceedings of the 1987 International Conference on Parallel Processing*, pages 630-637, St. Charles, Illinois, August 1987.

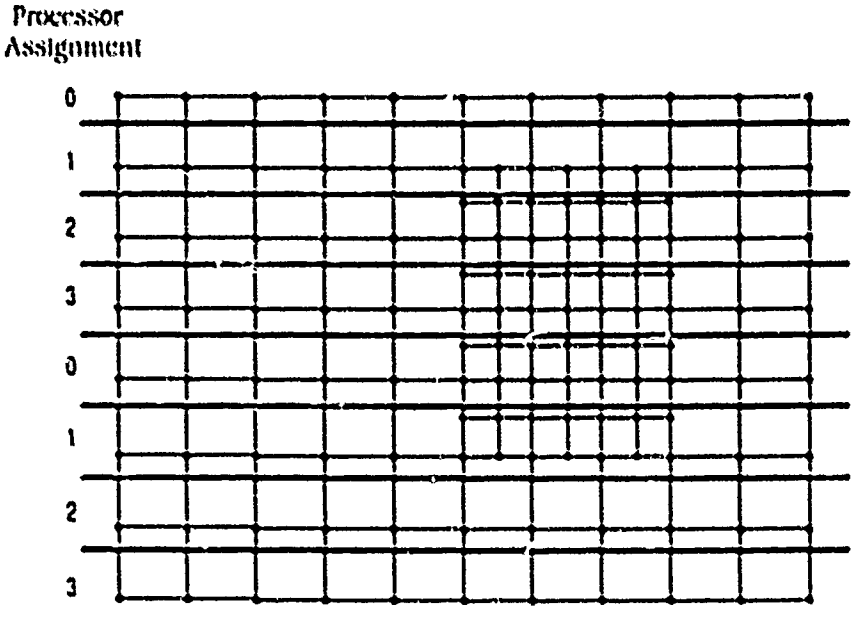


Figure 1: Scatter decomposition of an irregular grid

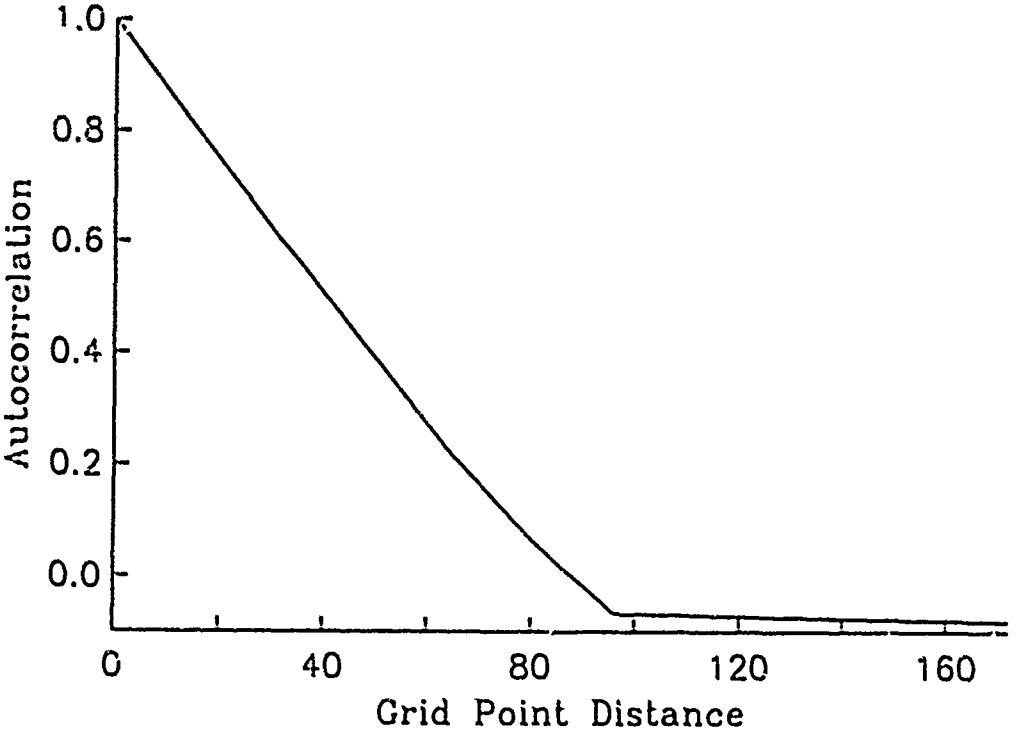


Figure 2: Correlation as function of distance in 1D fluids problem

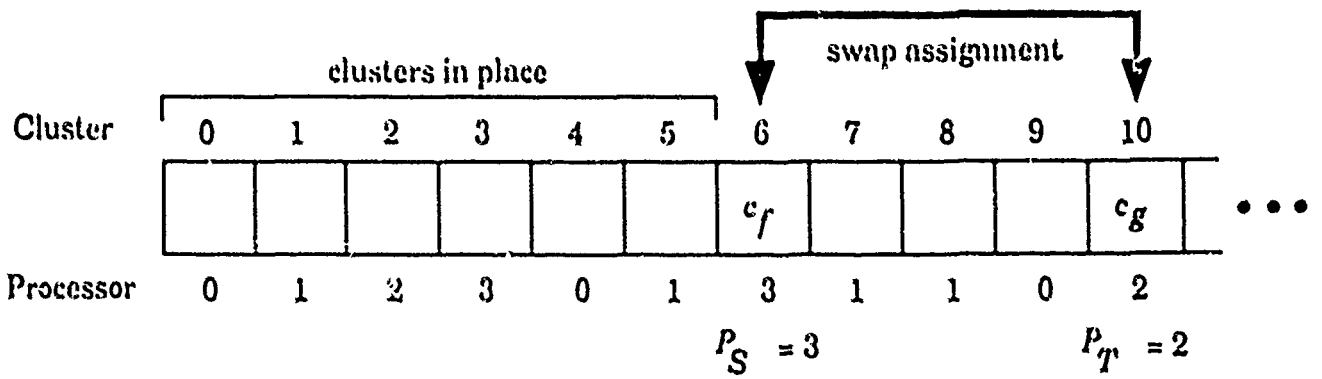


Figure 3: Clusters swapped in variance minimization argument

NASA

# Report Documentation Page

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA CR-181978<br>ICASE Report No. 90-4 | | |

**4. Title and Subtitle**

AN ANALYSIS OF SCATTER DECOMPOSITION

**5. Report Date**

January 1990

**6. Performing Organization Code**

**7. Author(s)**

David M. Nicol
Joel H. Saltz

**8. Performing Organization Report No.**

90-4

**10. Work Unit No.**

505-90-21-01

**9. Performing Organization Name and Address**

Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23665-5225

**11. Contract or Grant No.**

NAS1-18107
NAS1-18605

**13. Type of Report and Period Covered**

Contractor Report

**12. Sponsoring Agency Name and Address**

National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

**14. Sponsoring Agency Code**

**15. Supplementary Notes**

Langley Technical Monitor:
Richard W. Barnwell

Submitted to IEEE Trans. on Computers

Final Report

**16. Abstract**

This paper provides a formal analysis of a powerful mapping technique known as scatter decomposition. Scatter decomposition divides an irregular computational domain into a large number of equal sized pieces, and distributes them modularly among processors. We use a probabilistic model of workload in one dimension to formally explain why, and when scatter decomposition works. Our first result is that if correlation in workload is a convex function of distance, then scattering a more finely decomposed domain yields a lower average processor workload variance. Our second result shows that if the workload process is stationary Gaussian and the correlation function decreases linearly in distance until becoming zero and then remains zero, scattering a more finely decomposed domain yields a lower expected maximum processor workload. Finally we show that if the correlation function decreases linearly across the entire domain, then among all mappings that assign an equal number of domain pieces to each processor, scatter decomposition minimizes the average processor workload variance. The dependence of these results on the assumption of decreasing correlation is illustrated with situations where a coarser granularity actually achieves better load balance.

| 17. Key Words (Suggested by Author(s)) | 18. Distribution Statement |
|---|---|
| scatter decomposition; parallel processing; mapping algorithms; [illegible] | 66 - Systems Analysis<br>Unclassified - Unlimited |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 20 | A03 |

NASA FORM 1626 OCT 86